

Superposition as memory: unlocking quantum automatic complexity

Bjørn Kjos-Hanssen*

March 16, 2017

Abstract

We define the semi-classical quantum automatic complexity $Q_s(x)$ of a word x as the infimum in lexicographic order of those pairs of nonnegative integers (n, q) such that there is a subgroup G of the projective unitary group $\text{PU}(n)$ with $|G| \leq q$ and with $U_0, U_1 \in G$ such that, in terms of a standard basis $\{e_k\}$ and with $U_z = \prod_k U_{z(k)}$, we have $U_x e_1 = e_2$ and $U_y e_1 \neq e_2$ for all $y \neq x$ with $|y| = |x|$. We show that Q_s is unbounded and not constant for strings of a given length. In particular,

$$Q_s(0^2 1^2) \leq (2, 12) < (3, 1) \leq Q_s(0^{60} 1^{60})$$

$$\text{and } Q_s(0^{120}) \leq (2, 121).$$

1 Introduction

Quantum locks. Imagine a lock with two states, “locked” and “unlocked”, which may be manipulated using two operations, called 0 and 1. Moreover, the only way to (with certainty) unlock using four operations is to do them in the sequence 0011, i.e., $0^n 1^n$ where $n = 2$. In this scenario one might think that the lock needs to be in certain further states after each operation, so that there is some memory of what has been done so far. Here we show that this memory can be entirely encoded in superpositions of the two basic states “locked” and “unlocked”, where, as dictated by quantum mechanics, the operations are given by unitary matrices. Moreover, we show using the Jordan–Schur lemma that a similar lock is not possible for $n = 60$.

Quantum security. A problem with traditional padlocks is that a clever lock-breaker can seek to detect what internal state the lock is in part-way through the entering of the lock code. This problem disappears when the internal states

*This work was partially supported by a grant from the Simons Foundation (#315188 to Bjørn Kjos-Hanssen). This material is based upon work supported by the National Science Foundation under Grant No. 1545707.

are just superpositions of “locked” and “unlocked”. Of course, there may be a positive probability that the system when observed part-way through the entering of the lock code is observed in the “unlocked” state. To remedy this, one could use a sequence of many locks, say 10^k for a suitable positive integer k , and add a third “permanently locked” state which, once reached, cannot be left. Then observing the lock will likely eventually result in landing in the permanently locked state. (In the case of the code $0^n 1^n$, the permanently locked state might be implemented as having a probability related to the difference in the number of 0s and 1s entered in the code so far.) Note that in theory this is a purely quantum phenomenon: any simulation of the quantum device using classical hardware will be subject to the original problem that a lock-breaker may try to discern the internal states of the classical hardware.

Quantum automata. One of the fascinating aspects of quantum mechanics is how our understanding of states is enriched, with observable states, pure states, and mixed states. The notion of *state* of a finite automaton begs for a generalization to the quantum realm. Indeed, quantum finite automata have been studied already [1].

On the other hand automatic complexity introduced by Shallit and Wang [2] has been related it to model selection in statistics [3] and to pseudorandomness generation with linear feedback shift registers [4]. Other approaches to automatic complexity [5, 6, 7] yield better insight into infinite words.

We shall consider complexity with respect to an arbitrary semigroup before considering the quantum case of the projective unitary group $\text{PU}(n)$.

Definition 1. Let T_X denote the set of all transformations of the set X ; $T_X = \{f \mid f : X \rightarrow X\}$. The complexity of a string $x \in \{0, 1\}^n$, $n \geq 0$, is the class of all semigroup actions $\varphi : G \rightarrow T_X$ for semigroups G and sets X , with

- two¹ elements $\delta_0, \delta_1 \in G$, inducing $\delta_y = \prod_{k=1}^{|y|} \delta_{y(k)}$ for each $y \in \{0, 1\}^n$, $y = y(1) \cdots y(|y|)$;
- an initial state $\alpha \in X$; and
- a final state $\omega \in X$,

such that x is the only $y \in \{0, 1\}^n$ for which $\delta_y \alpha := \varphi(\delta_y) \alpha = \omega$.

In this case we say that x has complexity at most φ , or, if φ is understood, complexity at most G .

1.1 Quantum automatic complexity

Let $e_j^{(n)}$, $1 \leq j \leq n$ be the standard basis for \mathbb{C}^n . Let $\text{U}(n)$ be the group of unitary complex $n \times n$ matrices and let $\text{PU}(n)$ be the projective unitary group.

¹noncommuting, unless x is a unary string like 0^n

For $n \times n$ matrices U_0 and U_1 and a binary string x , we define

$$U_x = \prod_{k=1}^{|x|} U_{x(k)}.$$

Definition 2. A quantum deterministic finite automaton (*quantum DFA*) M with q states consists of an initial state $\alpha \in \mathbb{CP}^q$, a final state ω , and $\delta_0, \delta_1 \in \text{PU}(q)$. We say that M accepts a word $x \in \{0, 1\}^n$, $n \geq 0$ if

$$\delta_x \alpha = \omega.$$

Let $x \in \{0, 1\}^n$, $n \geq 0$. The quantum automatic complexity of x , $Q(x)$, is the least q such that there exists a quantum DFA M with q states such that for all $y \in \{0, 1\}^n$, M accepts y iff $y = x$.

- If we additionally require that δ_0, δ_1 generate a finite subgroup of $\text{PU}(q)$, we obtain the finite quantum automatic complexity $Q_f(x)$.
- If we require $\alpha = e_1$ and $\beta = e_2$ then we obtain semi-classical quantum automatic complexity Q_s .
- If we require both of the extra requirements for Q_s and Q_f , we get Q_{sf} .

We can write $Q_s(x) \leq (n, \infty)$ if $Q_s(x) \leq n$, and $Q_s(x) \leq (n, f)$ if $Q_{sf}(x) \leq n$ as witnessed by a finite group of order f . This way we see A_{perm} , the automatic complexity [2] with the added restriction that the transition functions be permutations, as an upper bound for n and a lower bound for f . Ordering these pairs lexicographically, we shall show that

$$(3, 121) \leq Q_s(0^{60}1^{60}) \leq (121, \infty)$$

assuming the following conjecture.

Conjecture 3. $A_{\text{perm}}(x) = |x| + 1$ for all x .

Remark 4. We have verified Conjecture 3 for binary strings of length up to 9.

Theorem 5. (1) If $Q_s(x) > (n, \infty)$ then $Q_s(x) \geq (n + 1, A_{\text{perm}}(x))$. (2) We always have $Q_s(x) \leq (A_{\text{perm}}(x), A_{\text{perm}}(x))$.

Proof. For (1) we note that the quantum states can be considered as states, so that the Cayley graph of any group witnessing Q_{sf} can be thought of as a witness for A_{perm} . For (2) we note that we can restrict attention to only the states $e_j^{(q)}$, $1 \leq j \leq q$, refusing to use superposition. \square

Matrices M of dimension $n \times n$ whose entries are 0 and 1, with exactly one 1 per column, act on $X = [n] = \{1, \dots, n\}$ by matrix multiplication in the following way:

$$\varphi(M)(j) = k, \quad \text{where } M e_j^{(n)} = e_k^{(n)}.$$

If M is additionally invertible then it thus induces an element of the symmetric group S_n and belongs to $O(n)$, the group of orthogonal matrices M (satisfying $M^{-1} = M^T$).

Theorem 6. *For each string x , $Q_f(x)$ is finite.*

Proof. By the embedding of S_n into $O(n)$ above, and then inclusion of $O(n)$ into $U(n)$ (simply because $U^{-1} = U^T$ for a real matrix U implies $U^{-1} = U^\dagger$), we have $Q_f(x) \leq (A_{\text{perm}}(x), A_{\text{perm}}(x)) \leq (x+1, x+1)$. \square

We also have $Q \leq Q_f \leq Q_{sf}$ and $Q \leq Q_s \leq Q_{sf}$. For our quantum lock analogy we want distinct initial and final states, whereas for automatic complexity $A(x)$ or $A_{\text{perm}}(x)$ it is natural to not require that.

2 Bounds on Q_s

Arbitrary q -state DFA transition functions δ_0, δ_1 can be considered to belong to the matrix algebra M_q of all $q \times q$ matrices. They are then exactly the matrices whose entries are 0 and 1, with exactly one 1 per column. And the nondeterministic case just corresponds to 0–1 valued matrices with *not necessarily* exactly one 1 per column. Moving to arbitrary real matrices we can significantly reduce the required dimension, from $n/2 + 1$ [8] to 2, as we now explain.

The following Theorem 7 indicates how any binary string can be encoded, in a sense, by two 2×2 matrices.

Theorem 7. *For each binary string x there exist $U_0, U_1 \in \text{GL}_2(\mathbb{R})$ such that $U_x e_1 = e_2$ and for any $y \neq x$, $|y| = |x|$, $U_y e_1 \neq e_2$.*

We omit the proof.

Theorem 8. *$Q(x) \leq 2$ for all strings x .*

Proof. It suffices to show that there is a free group generated by two unitary matrices. It is well-known [9] that a generic pair of unitaries in $U(2)$ generates a free group. Indeed, the existence of free subgroups of $SO(3)$ (and hence its double cover $SU(2)$) was already known to F. Hausdorff [10]; see also [11] and explicit examples in [12]. \square

Unfortunately, perhaps, free groups are incompatible with the “semi-classical” $e_1 \mapsto e_2$ property in the following way:

Theorem 9. *There is no word x of length > 0 and pair of unitary matrices U_0, U_1 such that U_0 and U_1 generate a free group and $U_x e_1 = e_2$ in projective space.*

3 Unboundedness of Q_f

As usual we denote by $H \trianglelefteq G$ that H is a normal subgroup G , and by $[G : H]$ the index of H in G .

Theorem 10 (Jordan–Schur). *There is a function $f(n)$ such that given a finite group G that is a subgroup of $M_n(\mathbb{C})$, there is an abelian subgroup $H \trianglelefteq G$ such that $[G : H] \leq f(n)$.*

	$e_1 \mapsto e_2$ required	not required
finite group required	∞	∞
not required	unknown	2

Table 1: Supremum of quantum automatic complexity over all strings. In the case where $e_1 \mapsto e_2$ is required (semi-classical quantum automatic complexity Q_s) but finiteness (Q_f) is not, we at least know that free groups cannot answer the question, by Theorem 9.

Corollary 11. *For each n there exists an m such that for any $u, v \in U(n)$ which generate a finite group, we have $[u^m, v^m] = 1$, i.e., $u^m v^m = v^m u^m$.*

Proof of Corollary from Theorem. If G is a finite group generated by u and v , and H a normal abelian subgroup of index $[G : H] = m$, then $u^m H = H$ and $v^m H = H$ (since any group element raised to the order of the group is the identity) and so u^m and v^m belong to H , hence, H being abelian, they commute. \square

Theorem 12. *For each n there is a binary string x with $Q_f(x) > n$.*

Proof. Let $x = 0^m 1^m$ where m is as in Corollary 11. Given $\delta_0, \delta_1 \in \text{PU}(n) = U(n)/U(1)$, choose x and y in $U(n)$ such that $\delta_0 = xU(1)$ and $\delta_1 = yU(1)$. Then $\delta_{0^m 1^m} = (xU(1))^m (yU(1))^m = x^m y^m U(1) = y^m x^m U(1) = \delta_{1^m 0^m}$. \square

The extent to which 2×2 matrices suffice for quantum automatic complexity is indicated in Table 1.

Theorem 13. $Q_{sf}(0^{60}1^{60}) > 2$.

Proof. Note that we may assume our finite subgroups are primitive as there is no point in having a separate automaton disconnected from the witnessing one. Collins [13] then shows that for $n = 2$, the optimal value is $m = 60$. \square

On the other hand, we show below in Theorem 19 that $Q_{sf}(0^{21}1^2) = 2$, leaving a gap $(2, 60)$ for the least n such that $Q_{sf}(0^n 1^n) > 2$. The state of our knowledge of finiteness of quantum automatic complexity is given in Table 1.

4 Calculating $Q_s(0011) \leq (2, 12)$

The group $\text{SU}(2)$ is the group of unit quaternions with the matrix representation [14]

$$\mathbf{1} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \mathbf{i} = \begin{bmatrix} i & 0 \\ 0 & -i \end{bmatrix}, \quad \mathbf{j} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}, \quad \mathbf{k} = \begin{bmatrix} 0 & i \\ i & 0 \end{bmatrix}$$

where i is the imaginary unit. We shall consider its order 24 subgroup the binary tetrahedral group

$$\left\{ \pm \mathbf{1}, \pm \mathbf{i}, \pm \mathbf{j}, \pm \mathbf{k}, \frac{1}{2}(\pm \mathbf{1} \pm \mathbf{i} \pm \mathbf{j} \pm \mathbf{k}) \right\},$$

also known by isomorphism as $\text{SL}(2, 3)$. Moreover we shall consider the order 12 quotient $\text{PSL}(2, 3)$ which is isomorphic to the alternating group $\text{Alt}(4)$.

Theorem 14. *There exist $\mathbf{a}, \mathbf{b} \in \text{SU}(2)$ such that*

$$\mathbf{aabb} \notin \{\mathbf{a}, \mathbf{b}\}^4 \setminus \{\mathbf{aabb}\}.$$

Proof. It turns out we can use the binary tetrahedral group to realize 0011 within $\text{SU}(2)$. Namely, let

$$\mathbf{a} = \delta_0 = (\mathbf{1} + \mathbf{i} + \mathbf{j} - \mathbf{k})/2, \quad \mathbf{b} = \delta_1 = (\mathbf{1} + \mathbf{i} + \mathbf{j} + \mathbf{k})/2$$

in the quaternion representation,

$$\mathbf{a} = \frac{1}{2} \begin{bmatrix} 1+i & 1-i \\ -i-1 & 1-i \end{bmatrix}, \quad \mathbf{b} = \frac{1}{2} \begin{bmatrix} 1+i & 1+i \\ i-1 & 1-i \end{bmatrix}.$$

We can check that $\mathbf{aabb} = -\mathbf{j}$ is unique among 4-letter words in \mathbf{a}, \mathbf{b} . \square

Theorem 15. *For $x = 0011$, there exist $\delta_0, \delta_1 \in \text{SO}(3)$ such that for all $y \in \{0, 1\}^4$, $\delta_y = \delta_x$ iff $y = x$.*

Proof. Another way to express \mathbf{a} and \mathbf{b} in Theorem 14 is as

$$e^{i\varphi} \begin{bmatrix} e^{i\Psi} & 0 \\ 0 & e^{-i\Psi} \end{bmatrix} \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} e^{i\Delta} & 0 \\ 0 & e^{-i\Delta} \end{bmatrix}$$

where $\varphi = 0$, $\theta = \pi/4$, and \mathbf{a} has $(\Psi, \Delta) = (0, \pi/4)$ and \mathbf{b} has $(\Psi, \Delta) = (\pi/4, 0)$. Thus

$$\mathbf{a} = \frac{1-i}{2} \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} i & 0 \\ 0 & 1 \end{bmatrix} = \left(\frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix} \right) \left(\frac{1-i}{\sqrt{2}} \begin{bmatrix} i & 0 \\ 0 & 1 \end{bmatrix} \right)$$

is a product of two matrices \mathbf{rs} in $\text{SU}(2)$. The first one, \mathbf{r} , corresponds [15] to the $\text{SO}(3)$ rotation

$$\begin{bmatrix} 0 & 0 & -1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

which is a 90-degree rotation in the xz -plane, and the second one, \mathbf{s} , to a 90-degree rotation

$$\begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

in the xy -plane in $\text{SO}(3)$. We have

$$\mathbf{b} = \frac{1-i}{2} \begin{bmatrix} i & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix} = \mathbf{sr}.$$

\square

One remaining wrinkle, taken care of in Theorem 19, is to make sure the other words are not only distinct from **aabb**, but map the start state to distinct vectors from what **aabb** does.

Theorem 16 (well known). *The order 24 group $\text{SL}(2, 3)$ is given by $a^3 = b^3 = c^2 = abc$, or equivalently $a^3 = b^3 = abab$.*

Theorem 17 ([16]). *$\text{SL}(2, 3)$ is isomorphic to the binary tetrahedral group, a subgroup of $\text{U}(2)$.*

The group $\text{Alt}(4)$ does serve as complexity bound for 0011. It is not a subgroup of $\text{U}(2)$ [16], but:

Theorem 18. *There is a faithful, irreducible representation of $\text{Alt}(4) \cong \text{PSL}(2, 3)$ as a subgroup of $\text{SL}(2, 3)$ of index 2 and as a subgroup of $\text{PU}(2)$.*

Proof. Let a be as in Theorem 16. We define an equivalence relation \equiv by $u \equiv v \iff u \in \{v, a^3v\}$. It is required to show that each element of our $\text{SL}(2, 3)$ is equivalent to an element of $\text{Alt}(4)$. This is done in detail in Figure 1. \square

The representation from Theorem 18 is used in the proof of Theorem 19.

Theorem 19. $Q_{sf}(0011) = 2$.

Proof. Let $v = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}$ with $v_1, v_2 \in \mathbb{R}$. Let

$$E_0 = \mathbf{a} = \frac{1}{2} \begin{bmatrix} 1+i & 1-i \\ -1-i & 1-i \end{bmatrix}, \quad E_1 = \mathbf{b} = \frac{1}{2} \begin{bmatrix} 1+i & 1+i \\ -1+i & 1-i \end{bmatrix}.$$

Let

$$D = \begin{bmatrix} v_1 & -v_2 \\ v_2 & v_1 \end{bmatrix} = [v \mid E_{0011}v], \quad C = \frac{1}{\sqrt{\det D}} D.$$

Let

$$U_j = C^{-1} E_j C, \quad j \in \{0, 1\}.$$

Then it follows that

$$CU_{0011} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = E_{0011} C \begin{bmatrix} 1 \\ 0 \end{bmatrix} = E_{0011} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = C \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

Hence

$$U_{0011} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

Since fortunately our E_0 and E_1 satisfy $E_{0011} = -\mathbf{j}$, C is orthogonal, and in particular C is unitary. If we now choose $v = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$, then v is sufficiently generic

$$\begin{aligned}
1, & \quad (1) \\
a, & \quad (2) \\
b, & \quad (3) \\
a^2 &= \textcolor{red}{bab}, & (4) \\
ab, & \quad (5) \\
ba, & \quad (6) \\
b^2 &= \textcolor{red}{aba}, & (7) \\
a^3 &= b^3 = baba = abab \equiv \textcolor{brown}{1}, & (8) \\
\textcolor{red}{a^2b} &= bab^2, & (9) \\
\textcolor{red}{ab^2} &= \textcolor{blue}{a^2ba}, & (10) \\
\textcolor{red}{ba^2} &= \textcolor{blue}{b^2ab} = aba^2b, & (11) \\
\textcolor{red}{b^2a} &= \textcolor{blue}{aba^2} = ab^2ab, & (12) \\
a^4 &= ab^3 = b^3a = a^2bab = ababa \equiv \textcolor{brown}{a}, & (13) \\
a^3b &= ba^3 = b^4 = abab^2 \equiv \textcolor{brown}{b}, & (14) \\
\textcolor{blue}{a^2b^2} &= a^3ba \equiv \textcolor{brown}{ba}, & (15) \\
\textcolor{blue}{ab^2a} &= a^2ba^2, & (16) \\
\textcolor{blue}{ba^2b} &\equiv \textcolor{brown}{abba}, & (17) \\
\textcolor{blue}{b^2a^2} &= a^4b = aba^3 = ab^4 \equiv \textcolor{brown}{ab}, & (18) \\
a^5 &= a^2b^3 = ab^3a \equiv \textcolor{brown}{a^2}, & (19) \\
a^3b^2 &\equiv \textcolor{brown}{b^2}, & (20) \\
a^2b^2a &\equiv \textcolor{brown}{baa}, & (21) \\
ab^2a^2 &\equiv \textcolor{brown}{aab}, & (22) \\
ba^2b^2 &\equiv \textcolor{brown}{bba}, & (23) \\
b^2a^2b &\equiv \textcolor{brown}{abb}. & (24)
\end{aligned}$$

Figure 1: The 24 elements of $\text{SL}(2, 3)$. All strings of length at most 2 are unique of their length. By symmetry, words of length 5 starting with b are not written down.

that $U_y e_1 \neq e_2$ as elements of \mathbb{CP}^1 for all $y \in \{0, 1\}^4 \setminus \{x\}$. We have verified as much with an Octave computation (see Figure 2 and Figure 3). We have

$$C = \frac{1}{\sqrt{5}} \begin{bmatrix} 1 & -2 \\ 2 & 1 \end{bmatrix}, \quad C^{-1} = \frac{1}{\sqrt{5}} \begin{bmatrix} 1 & 2 \\ -2 & 1 \end{bmatrix},$$

$$U_0 = \frac{1}{10} \begin{bmatrix} 5+i & 5+7i \\ -5+7i & 5-i \end{bmatrix} \quad \text{and} \quad U_1 = \frac{1}{10} \begin{bmatrix} 5-7i & 5+i \\ -5+i & 5+7i \end{bmatrix}.$$

□

Remark 20. *It is still a question what the nature of the “complexity” Q and Q_s are picking out is. If it is anything like A_{perm} it may be less than intuitive. However, there is some reason to believe that quantum automatic complexity is better than permutation automatic complexity at distinguishing strings of the same length. For permutation automatic complexity we do not know any example of strings of the same length having distinct complexity, but for quantum automatic complexity, $0^{60}1^{60}$ and 0^{120} form such an example. The latter has complexity at most 2 whereas the former does not (Theorem 13).*

References

- [1] A. Kondacs and J. Watrous, “On the power of quantum finite state automata,” in *38th Annual Symposium on Foundations of Computer Science, FOCS ’97, Miami Beach, Florida, USA, October 19-22, 1997*, 1997, pp. 66–75. [Online]. Available: <http://dx.doi.org/10.1109/SFCS.1997.646094>
- [2] J. Shallit and M.-W. Wang, “Automatic complexity of strings,” *J. Autom. Lang. Comb.*, vol. 6, no. 4, pp. 537–554, 2001, 2nd Workshop on Descriptive Complexity of Automata, Grammars and Related Structures (London, ON, 2000).
- [3] B. Kjos-Hanssen, “Few paths, fewer words: model selection with automatic structure functions,” *Experimental Mathematics*, 2018, conditionally accepted; arXiv 1608.01399.
- [4] B. Kjos-Hanssen, “Shift registers fool finite automata,” *ArXiv e-prints*, Jul. 2016.
- [5] A. Shen, “Automatic Kolmogorov complexity and normality revisited,” *ArXiv e-prints*, Jan. 2017.
- [6] V. Becher, O. Carton, and P. A. Heiber, “Normality and automata,” *J. Comput. System Sci.*, vol. 81, no. 8, pp. 1592–1613, 2015. [Online]. Available: <http://dx.doi.org/10.1016/j.jcss.2015.04.007>

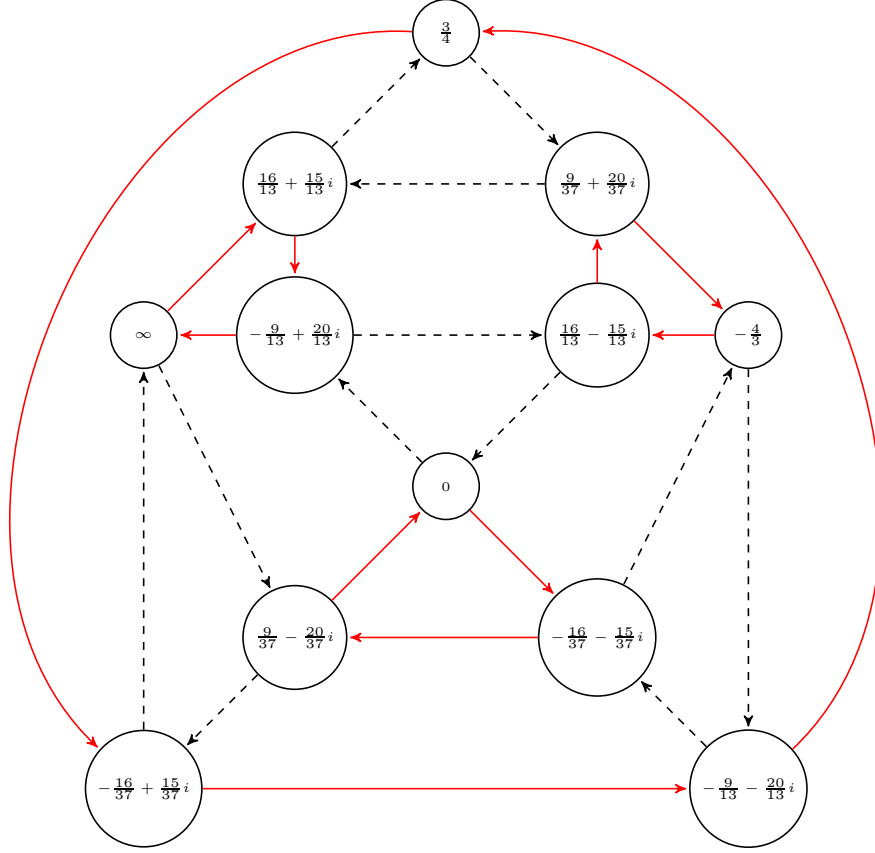


Figure 2: Quantum complexity witness having the shape of a cuboctahedron. The label α represents the projective point $[1 : \alpha]$. The initial state is $[1 : 0]$ denoted by 0 and the accept state is $[0 : 1]$ denoted by ∞ . Dashed lines indicate multiplication by U_0 . Solid lines indicate multiplication by U_1 .

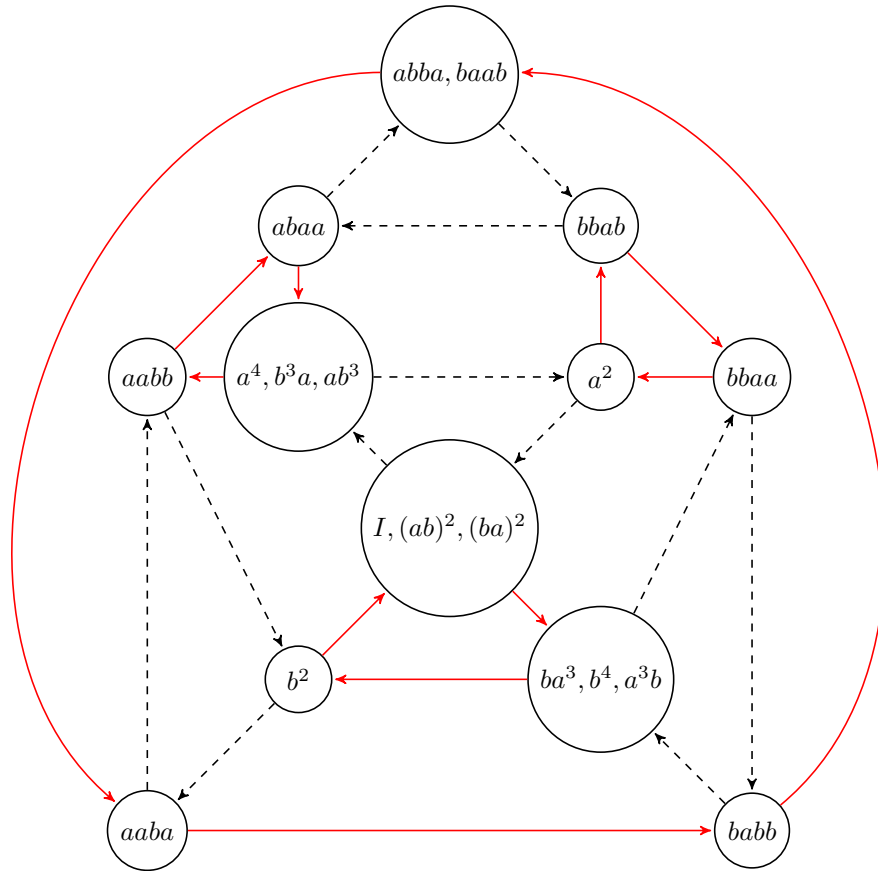


Figure 3: Another view of the quantum complexity witness having the shape of a cuboctahedron of Figure 2.

- [7] C. S. Calude, K. Salomaa, and T. K. Roblot, “Finite state complexity,” *Theoret. Comput. Sci.*, vol. 412, no. 41, pp. 5668–5677, 2011. [Online]. Available: <http://dx.doi.org/10.1016/j.tcs.2011.06.021>
- [8] K. Hyde and B. Kjos-Hanssen, “Nondeterministic automatic complexity of overlap-free and almost square-free words,” *Electron. J. Combin.*, vol. 22, no. 3, pp. Paper 3.22, 18 pp., 2015.
- [9] A. Thom, “Convergent sequences in discrete groups,” *Canad. Math. Bull.*, vol. 56, no. 2, pp. 424–433, 2013. [Online]. Available: <http://dx.doi.org/10.4153/CMB-2011-155-3>
- [10] F. Hausdorff, “Bemerkung über den Inhalt von Punktmengen,” *Math. Ann.*, vol. 75, no. 3, pp. 428–433, 1914. [Online]. Available: <http://dx.doi.org/10.1007/BF01563735>
- [11] S. Świerczkowski, “On a free group of rotations of the Euclidean space,” *Nederl. Akad. Wetensch. Proc. Ser. A 61 = Indag. Math.*, vol. 20, pp. 376–378, 1958.
- [12] —, “A class of free rotation groups,” *Indag. Math. (N.S.)*, vol. 5, no. 2, pp. 221–226, 1994. [Online]. Available: [http://dx.doi.org/10.1016/0019-3577\(94\)90026-4](http://dx.doi.org/10.1016/0019-3577(94)90026-4)
- [13] M. J. Collins, “On Jordan’s theorem for complex linear groups,” *J. Group Theory*, vol. 10, no. 4, pp. 411–423, 2007. [Online]. Available: <http://dx.doi.org/10.1515/JGT.2007.032>
- [14] Q. Yuan. (2011, 2) $su(2)$ and the quaternions. [Online]. Available: <https://qchu.wordpress.com/2011/02/12/su2-and-the-quaternions/>
- [15] I. M. Gelfand, R. A. Minlos, and Z. J. Sapiro, *Predstavleniya gruppy vrashcheni i gruppy Lorentsa, ikh primeneniya*. Gosudarstv. Izdat. Fiz.-Mat. Lit., Moscow, 1958.
- [16] K. M. Parattu and A. Wingerter. (2011) Tribimaximal mixing from small groups. [Online]. Available: <https://arxiv.org/pdf/1012.2842v2.pdf>